Spatial filtering/ Filtragem

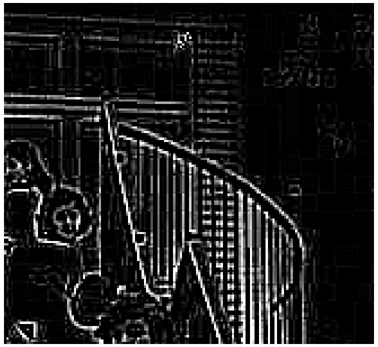
Filtros lineares

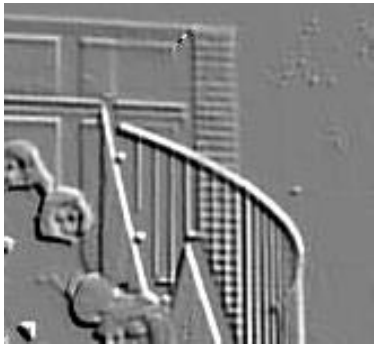
- Convolução
- Filtro passa-baixas (suavização)
- Filtro passa-altas (realce)
- Filtros direcionais
- Filtros não lineares

Processamento de imagens

Prof. Dr. Jorge Centeno

Para que servem?





Filtros lineares

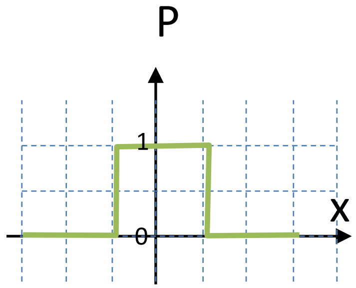
Os filtros lineares resultam da *convolução* de uma janela móvel e a imagem.

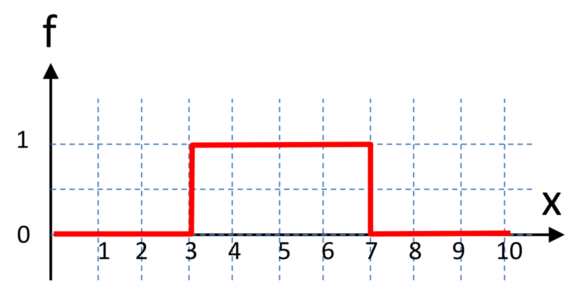
<u>A Convolução em uma dimensão</u>:

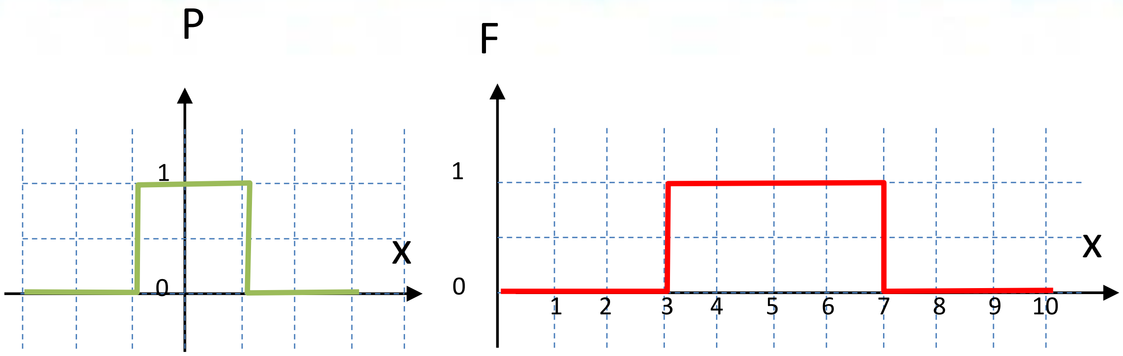
A convolução é um operador **linear** que, a partir de duas funções previamente definidas, resulta numa terceira função (resultado) definida ao longo do domínio das funções de entrada.

O operador da convolução mede a **soma do produto** das funções de entrada ao longo do domínio, definido pela superposição delas, deslocando uma delas em relação à outra.

Considere as duas funções discretas (1D) abaixo. Qual será o resultado da convolução? Como calcular?







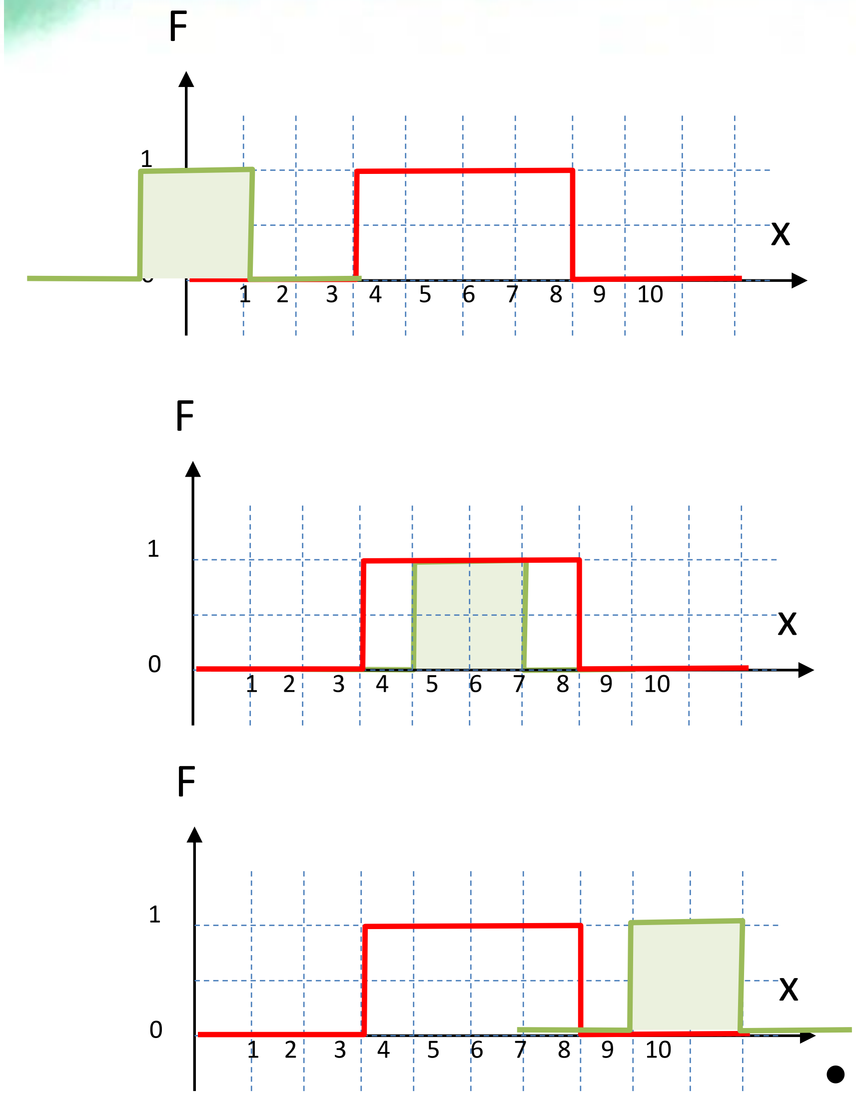
x	-2	-1	0	1	2	3	4	5	6	7	8	9	10
P	0	1	1	1	0								
F			0	0	0	1	1	1	1	1	0	0	0

Para uma dimensão a *convolução* de duas funções p(x) e F(x) é dada por:

$$G(x) = \Sigma \ F(x) * P(x+dx)$$

Aqui "dx" representa um deslocamento da segunda função em relação à origem.

Então, para cada possível "dx" deve-se calcular a soma da multiplicação das duas funções...



- deslocar

deslocar

x	-2	-1	0	1	2	3	4	5	6	7	8	9	10
P		0	1	1	1	0							
F			0	0	0	1	1	1	1	1	0	0	0

0

x	-2	-1	0	1	2	3	4	5	6	7	8	9	10
P			0	1	1	1	0						
F			0	0	0	1	1	1	1	1	0	0	0

1

x	-2	-1	0	1	2	3	4	5	6	7	8	9	10
P				0	1	1	1	0					
F			0	0	0	1	1	1	1	1	0	0	0

2

x	-2	-1	0	1	2	3	4	5	6	7	8	9	10
P					0	1	1	1	0				
F			0	0	0	1	1	1	1	1	0	0	0

3

x	-2	-1	0	1	2	3	4	5	6	7	8	9	10
P						0	1	1	1	0			
F			0	0	0	1	1	1	1	1	0	0	0

3

x	-2	-1	0	1	2	3	4	5	6	7	8	9	10
P						0	1	1	1	0			
F			0	0	0	1	1	1	1	1	0	0	0

3

• • •

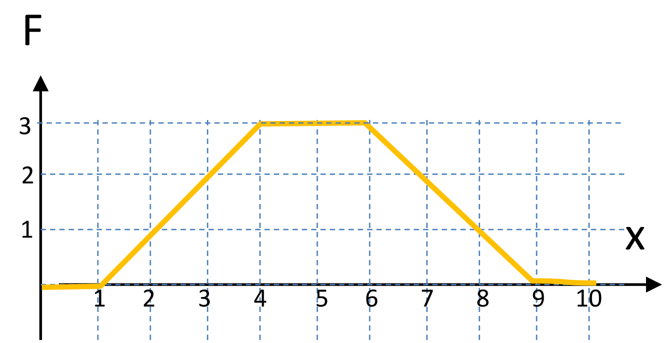
...

x	-2	-1	0	1	2	3	4	5	6	7	8	9	10
P									0	1	1	1	0
F			0	0	0	1	1	1	1	1	0	0	0

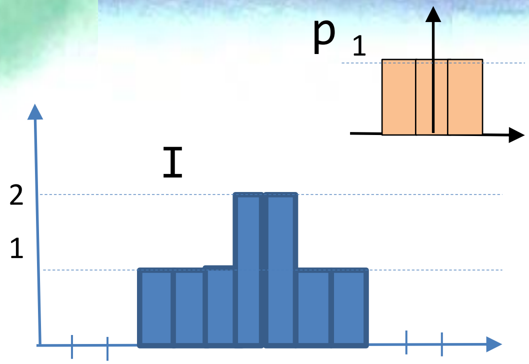
1

começando em 0

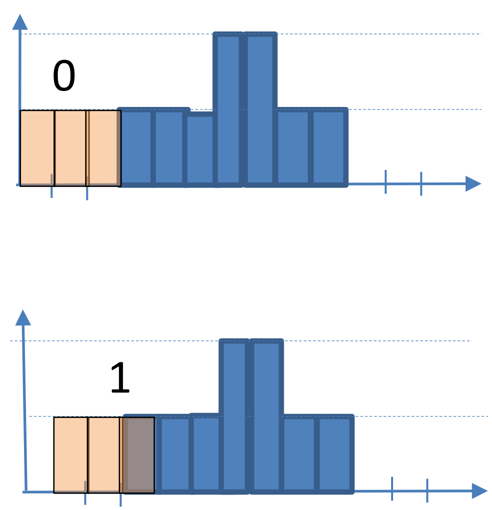
S	0	0	1	2	3	3	3	2	1	0

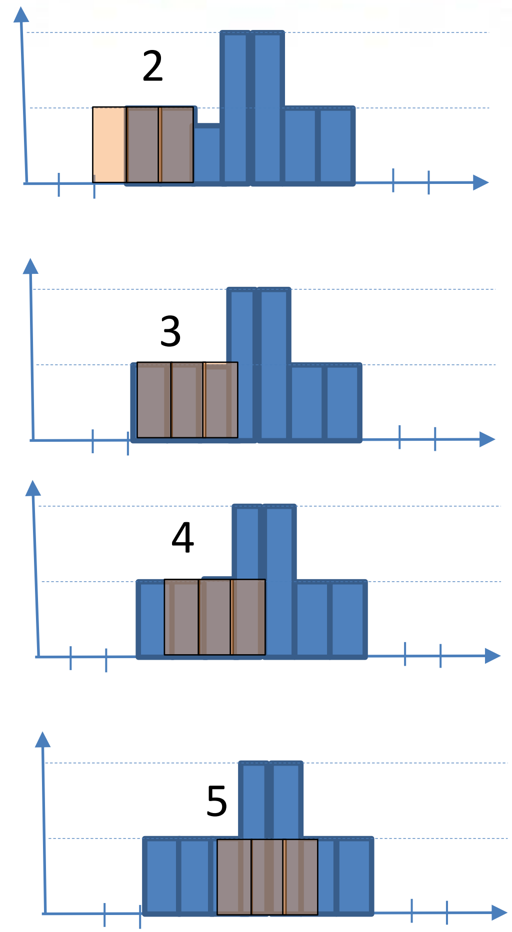


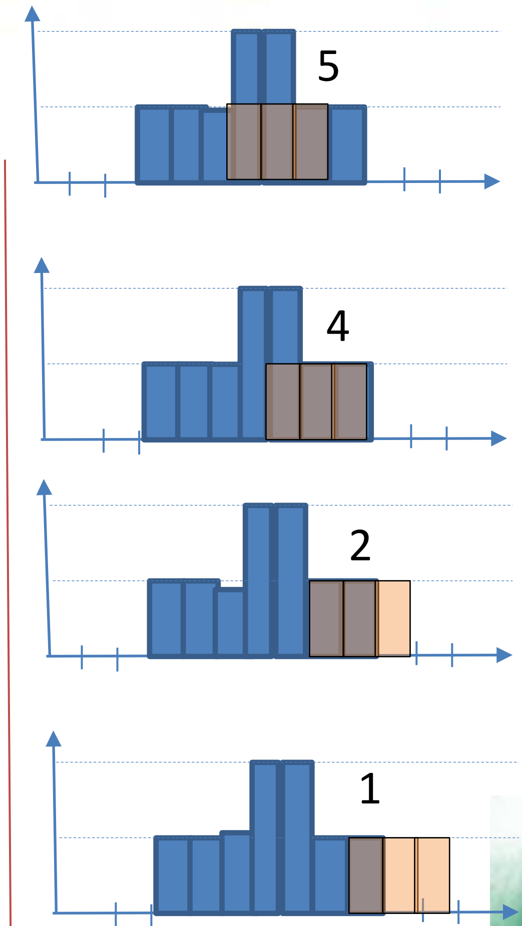
convolução



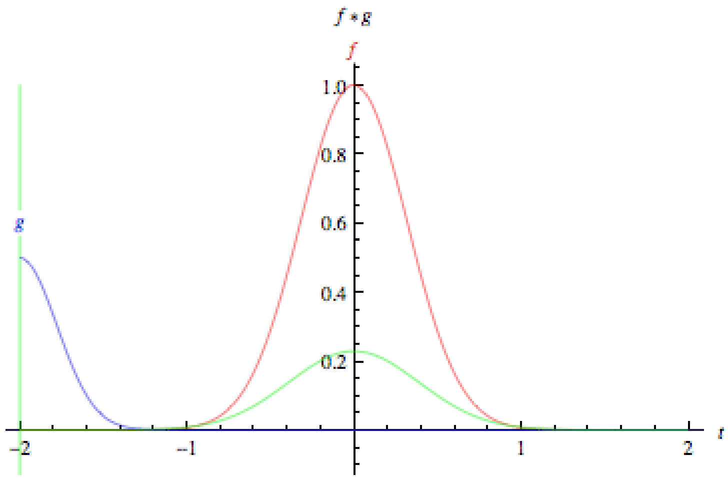
Deslocar “p”







Ao deslocar a segunda função e multiplicar pela primeira, são obtidos valores diferentes em função do deslocamento.



Duas propriedades

- A função que se desloca pode funcionar como um “filtro” que modifica a primeira função
- Variando a função que se desloca podem ser obtidos diferentes efeitos *filtros”
- Estas transformações são lineares

exemplo

Considerando uma função F(x), definida nos pontos x=1,2,3,...,12, com os seguintes valores:

x	1	2	3	4	5	6	7	8	9	10	11	12
F	5	2	5	5	5	9	5	5	4	4	5	5

e o filtro (P) de tamanho 1x3 com os valores

x	-1	0	1
P	1	1	1

Em cada instante deve-se deslocar o filtro (dx) operar (somar as respectivas multiplicações) e atribuir o resultado à posição considerada (dx)

Como não existem valores para x=0, o cálculo somente é possível para x=2

5 4 5 5 5 9 5 5 4 4 5 5

1 1 1

exemplo

P=1 5 2 5 5 5 9 5 5 4 4 5 5 . . . .

p=2 5 2 5 5 5 9 5 5 4 4 5 5 . 4 . . . .

p=3 5 2 5 5 5 9 5 5 4 4 5 5 . 4 4 . . . .

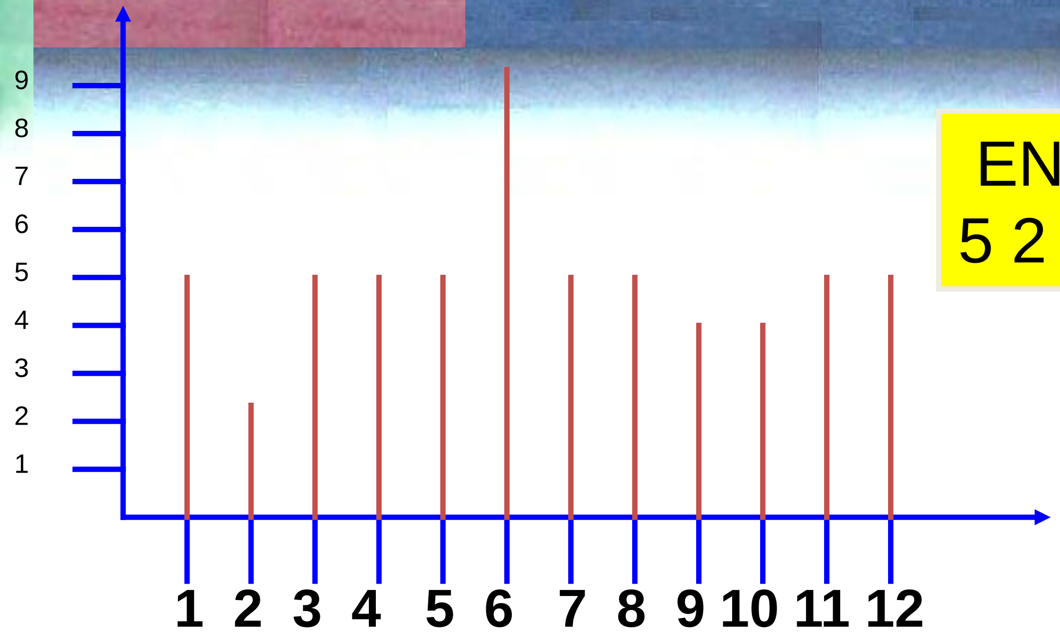
p=4 5 2 5 5 5 9 5 5 4 4 5 5 . 4 4 5 . . . .

5 2 5 5 5 9 5 5 4 4 5 5 . 4 4 5 6 . . . .

5 2 5 5 5 9 5 4 4 5 5 . 4 4 5 6 6 . . . .

p=12 . 4 4 5 6 6 6 5 4 4 5 .

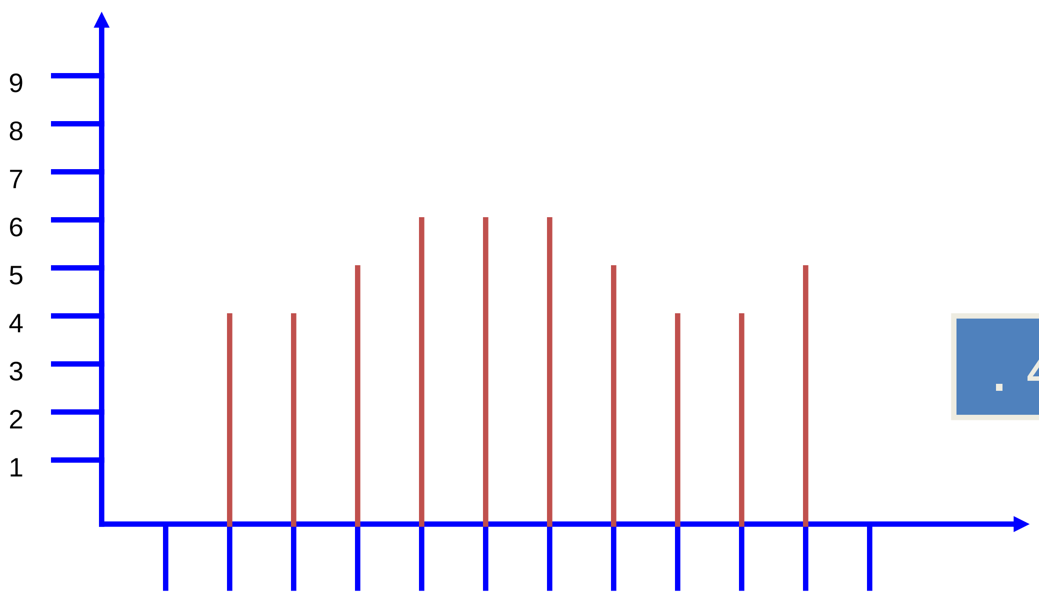
exemplo



ENTRADA
5 2 5 5 5 9 5 5 4 4 5 5

1 1 1

. 4 4 5 6 6 6 5 4 4 5 .



Filtros lineares

Em processamento de imagens, os filtros lineares resultam da convolução de uma janela móvel e a imagem no espaço 2-D.

O resultado de um filtro linear pode ser escrito na seguinte forma:

$$G(y,x) = \Sigma\ \Sigma\ (\ F(x,y) * p(x+dx, y+dy)\ )$$

onde

- y,x representam as coordenadas do pixel e
- dx, dy o deslocamento relativo
- p(i,j) representa o filtro
- G(y,x) é a imagem resultante, filtrada

1	1	1	1	2	1	1	0
1	1	2	1	3	1	1	1
1	1	3	9	2	1	1	0
1	1	3	1	2	1	1	1
9	7	8	9	8	9	9	8
9	9	8	8	9	9	9	8
9	9	9	8	9	8	9	9
9	8	9	8	8	9	9	9

1	1	1	1	2	1	1	0
1	1	2	1	3	1	1	1
1	1	3	9	2	1	1	0
1	1	3	1	2	1	1	1
9	7	8	9	8	9	9	8
9	9	8	8	9	9	9	8
9	9	9	8	9	8	9	9
9	8	9	8	8	9	9	9

1	1	1	1	2	1	1	0
1	1	2	1	3	1	1	1
1	1	3	9	2	1	1	0
1	1	3	1	2	1	1	1
9	7	8	9	8	9	9	8
9	9	8	8	9	9	9	8
9	9	9	8	9	8	9	9
9	8	9	8	8	9	9	9

1	1	1	1	2	1	1	0
1	1	2	1	3	1	1	1
1	1	3	9	2	1	1	0
1	1	3	1	2	1	1	1
9	7	8	9	8	9	9	8
9	9	8	8	9	9	9	8
9	9	9	8	9	8	9	9
9	8	9	8	8	9	9	9

1	1	1	1	2	1	1	0
1	1	2	1	3	1	1	1
1	1	3	9	2	1	1	0
1	1	3	1	2	1	1	1
9	7	8	9	8	9	9	8
9	9	8	8	9	9	9	8
9	9	9	8	9	8	9	9
9	8	9	8	8	9	9	9

1	1	1	1	2	1	1	0
1	1	2	1	3	1	1	1
1	1	3	9	2	1	1	0
1	1	3	1	2	1	1	1
9	7	8	9	8	9	9	8
9	9	8	8	9	9	9	8
9	9	9	8	9	8	9	9
9	8	9	8	8	9	9	9

1	1	1	1	2	1	1	0
1	1	2	1	3	1	1	1
1	1	3	9	2	1	1	0
1	1	3	1	2	1	1	1
9	7	8	9	8	9	9	8
9	9	8	8	9	9	9	8
9	9	9	8	9	8	9	9
9	8	9	8	8	9	9	9

1	1	1	1	2	1	1	0
1	1	2	1	3	1	1	1
1	1	3	9	2	1	1	0
1	1	3	1	2	1	1	1
9	7	8	9	8	9	9	8
9	9	8	8	9	9	9	8
9	9	9	8	9	8	9	9
9	8	9	8	8	9	9	9

3x3

Para um pixel na posição i,j, deve-se calcular o valor digital médio dele e seus oito vizinhos

- Os vizinhos são
- `[i-1,j-1] [i-1, j] [i-1,j+1]`
- `[i  ,j-1] [i  , j] [i , j+1]`
- `[i+1,j-1] [i+1, j] [i+1,j+1]`

Ou, variando um índice de -1 a 1 = dc=[-1,0, 1]

`[i-1,j+dc]; [i,j+dc]; [i+1,j+dc]`

E variando um índice de linhas `dl=[-1, 0,1]`

`[i+dl, j+dc]`

1	1	1	1	2	1	1	0
1	1	2	1	3	1	1	1
1	1	3	9	2	1	1	0
1	1	3	1	2	1	1	1
9	7	8	9	8	9	9	8
9	9	8	8	9	9	9	8
9	9	9	8	9	8	9	9
9	8	9	8	8	9	9	9

Filtro (3x3)

1 2 1
2 2 2
1 2 1

dividido por 14, a soma dos elementos:

1/14 2/14 1/14
2/14 2/14 1/14
1/14 2/14 1/14

Para a posição linha=5 coluna=3:
Considerando a matriz de ”pesos”.

G = 1/14*(1*1 +2*3 +1*1 +2*7 +2*8 +2*9+1*9+2*8+1*8)

= 89 /14

= 6,4

Ou em valor digital (arredondar, ou truncar)

= 6

1	1	1	1	2	1	1	0
1	1	2	1	3	1	1	1
1	1	3	9	2	1	1	0
1	1	3	1	2	1	1	1
9	7	8	9	8	9	9	8
9	9	8	8	9	9	9	8
9	9	9	8	9	8	9	9
9	8	9	8	8	9	9	9

Da definição de filtro, entende-se que:

- Um resultado pode ser calculado para cada posição da janela móvel dentro da imagem.
- O resultado é sempre atribuído ao pixel central.

Este novo valor é calculado somando o produto do pixel na janela 3x3 e o respectivo peso, definido no filtro.

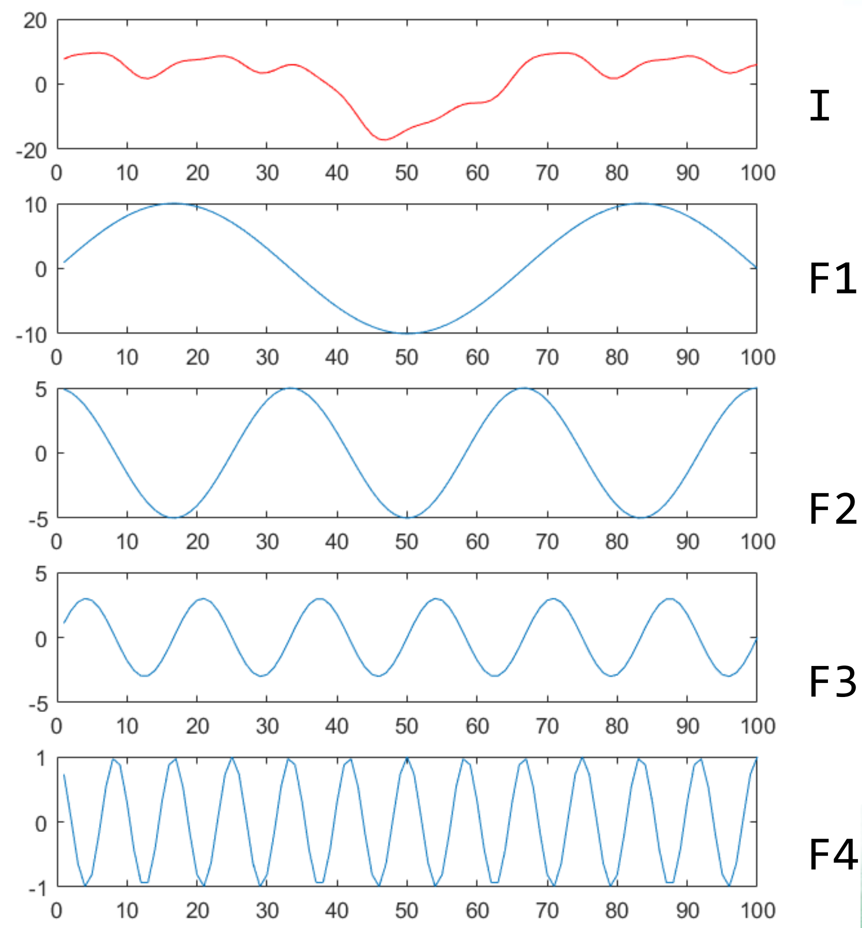
Os filtros lineares são definidos em termos de seu efeito na imagem como *passa baixas e passa altas*.

Conceito de Frequencias

Uma função pode ser descrita como a soma de várias funções senoidais com diferentes frequências.

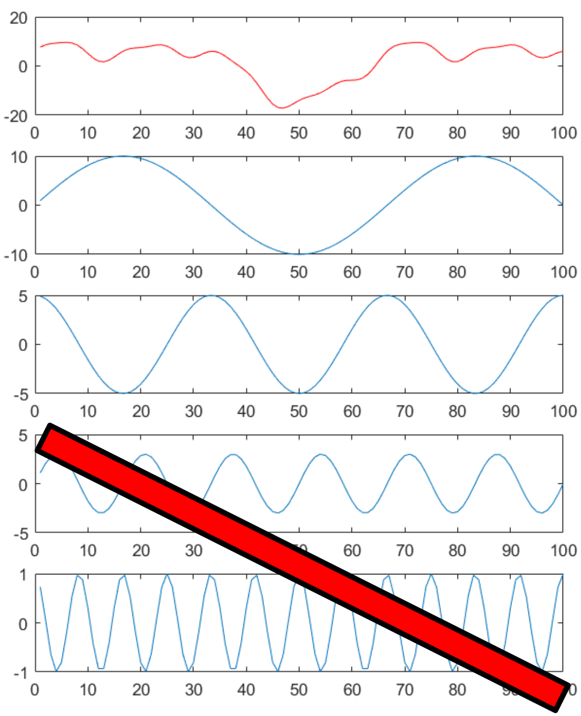
I=F1+F2+F3+F4 + ...

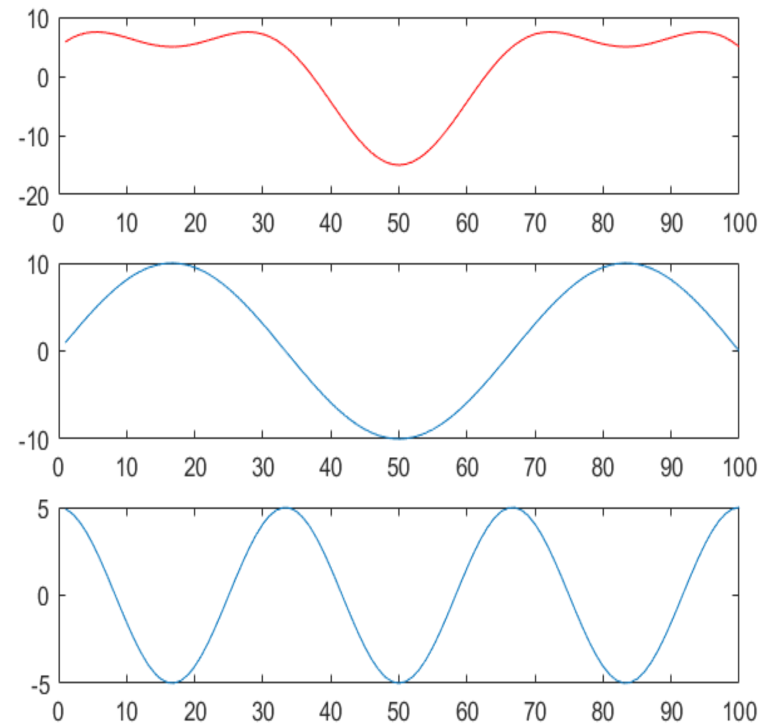
As funções com menores frequencias são responsáveis por dar a forma geral da curva, enquanto as com menores frequencias representam os detalhes



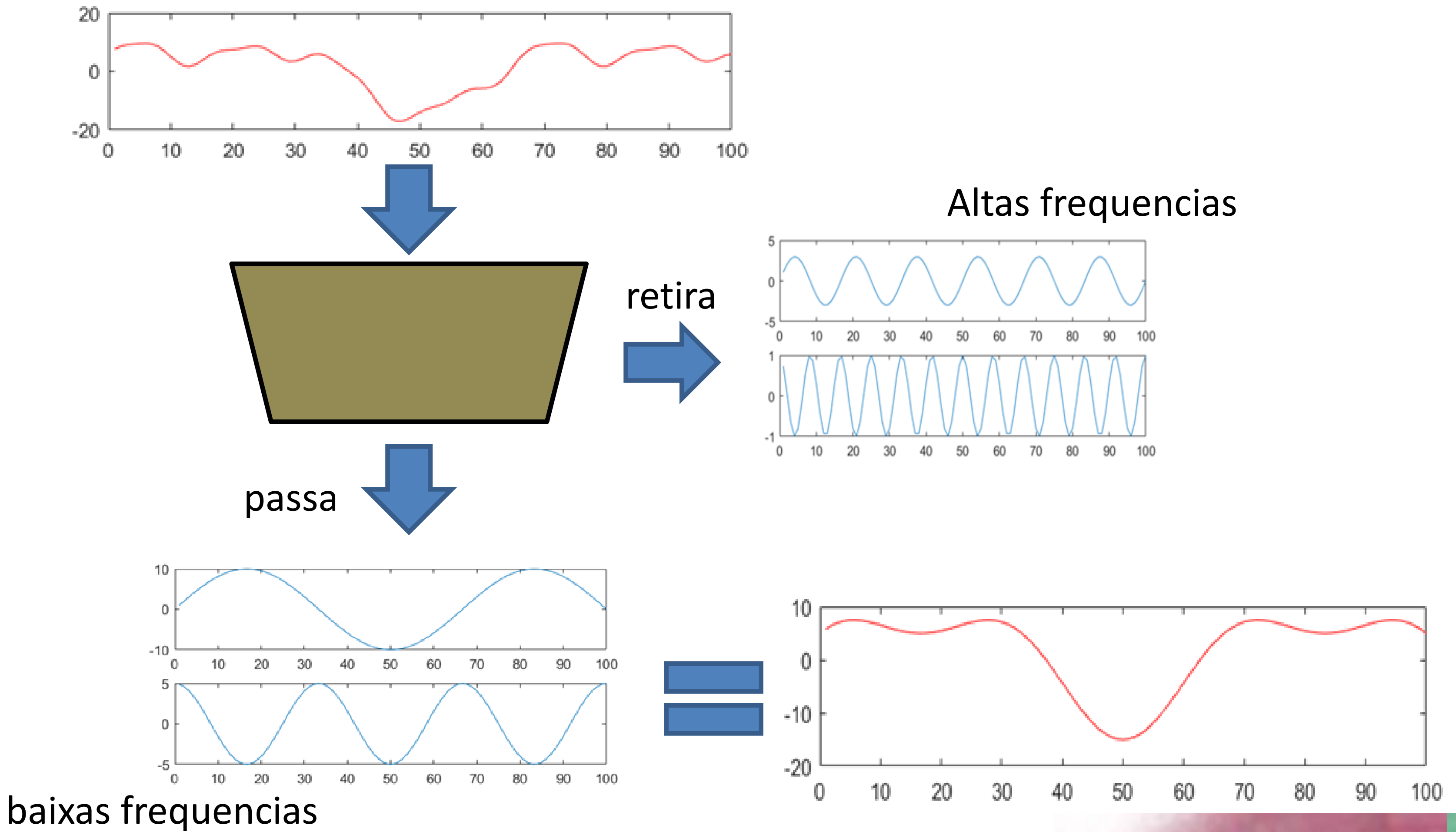
filtragem

O que ocorre se retiramos as altas frequencias?





Filtragem seletiva de frequencias



Filtro passa-baixas (suavização)

- Atenua as altas frequências, aquelas associadas a detalhes na imagem, e deixa apenas as baixas frequências. O efeito deste filtro é a remoção de detalhes da imagem e sua suavização. A imagem filtrada apresenta uma aparência de névoa ou um efeito de ``imagem fora de foco'', e as áreas presentes na imagem tornam-se mais homogêneas.
- O efeito é atingido substituindo o pixel central pela média da janela. A média pode ser uma média simples ou uma média ponderada, onde diferentes pesos são atribuídos aos vizinhos em função de sua proximidade ao pixel central.
- Exemplo:

1	1	1	1	2	1	1	0
1	1	2	1	3	1	1	1
1	1	3	9	2	1	1	0
1	1	3	1	2	1	1	1
9	7	8	9	8	9	9	8
9	9	8	8	9	9	9	8
9	9	9	8	9	8	9	9
9	8	9	8	8	9	9	9

Filtro (3x3)

1	**1**	**1**
1	**1**	**1**
1	**1**	**1**

Exemplos de filtros passa-baixas

1	1	1
1	1	1
1	1	1

1	2	1
2	4	2
1	2	1

1	1	1
1	4	1
1	1	1

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Filtros de média.
Para o cálculo do valor final, o resultado da multiplicação dos pesos e os valores da imagem deve ser dividido pela soma dos pesos.

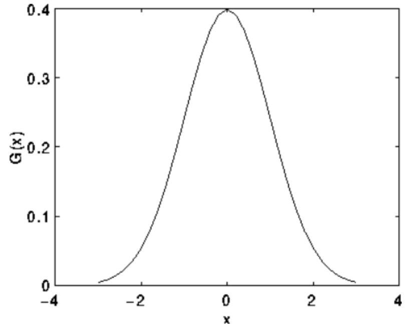
Passa baixas

- Original passa-baixas (suavização)

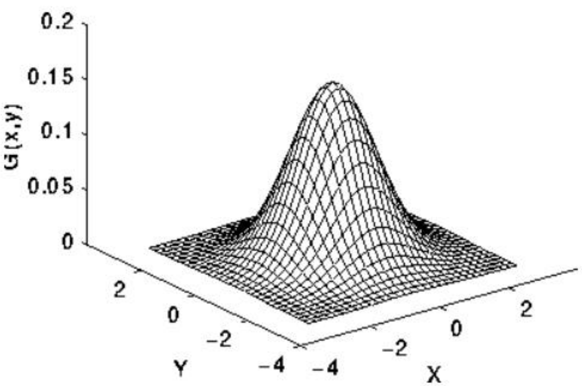
FILTRO GAUSSIANO

O Filtro Gaussiano é um tipo de filtro passa-baixas que usa uma função Gaussiana para calcular os pesos do filtro e, consequentemente, a transformação linear. Assim, maior peso é dado ao central e o peso diminui com a distancia ao pixel central da janela.

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$



$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

$$
\begin{aligned}
&= \sum_i \sum_j f(x-i, y-j) \exp\left\{\frac{-(i^2+j^2)}{2\sigma^2}\right\} \\
&= \sum_i \left[\sum_j f(x-i, y-j) \exp\left\{\frac{-j^2}{2\sigma^2}\right\}\right] \exp\left\{\frac{-i^2}{2\sigma^2}\right\} \\
&= [f(x,y) * G(y)] * G^T(x).
\end{aligned}
$$

Porém, um filtro Gaussiano 2D pode ser substituído por dois filtros Gaussianos 1D, que são mais rápidos.

- Exemplo

$$f * \left(\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \right) = \left(f * \frac{1}{4} [1 \quad 2 \quad 1] \right) * \left(\frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \right)$$

$$\frac{1}{4} [1 \quad 2 \quad 1] * \frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

Filtro passa-altas (realce)

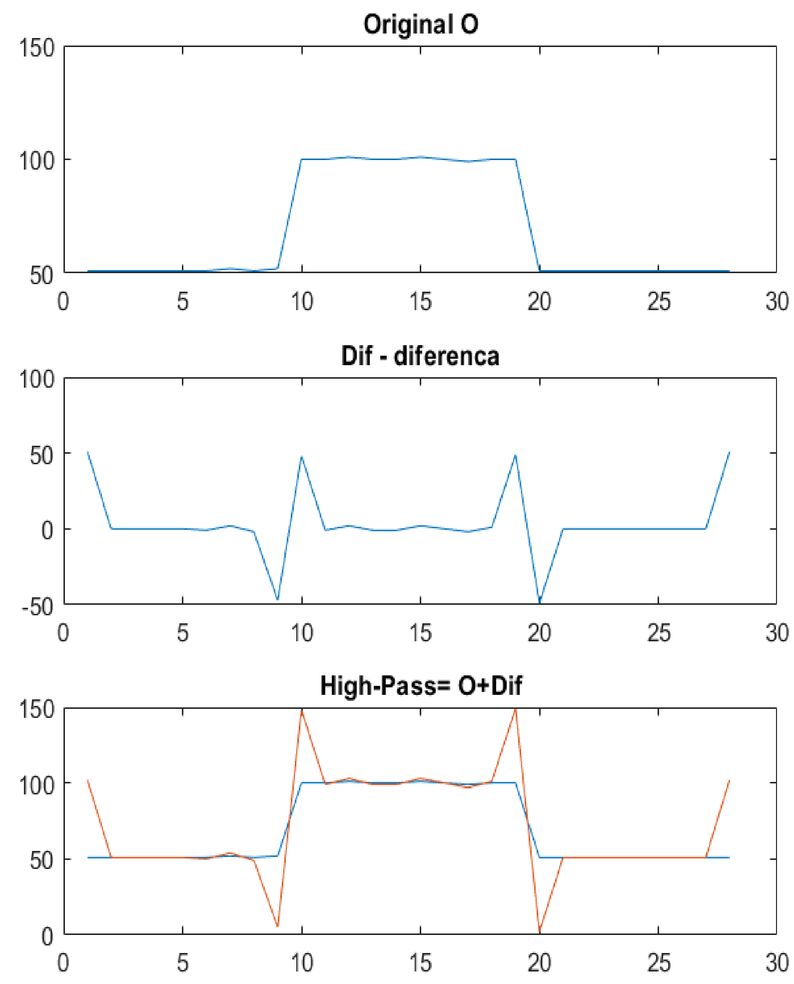
- Enfatiza os contrastes, realçando os detalhes da imagem.
- O nome do filtro explica seu funcionamento, pois nesta transformação as baixas frequências são eliminadas, sendo as altas frequências as únicas remanescentes.
- Este efeito pode ser atingido adicionando à imagem original a diferença entre a imagem original e o resultado de um filtro-passa baixas.
- O resultado da operação é nulo em regiões homogêneas. Em regiões com detalhes, o valor resultante é alto, em função do contraste entre o pixel central e a vizinhança.

Para entender...

Se aplicarmos a uma série de dados (1D)

A diferença retira o valor original, pode ser igual em regiões claras e escuras.

Somando esta diferença ao valor original se salienta o contraste nas regiões de bordas.
Em regiões uniformes, não ocorre alteração.

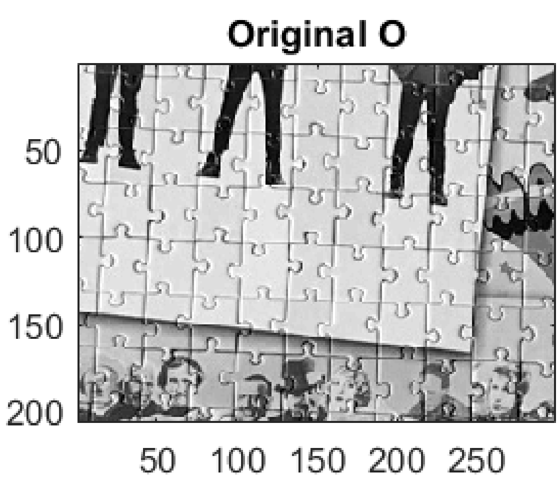


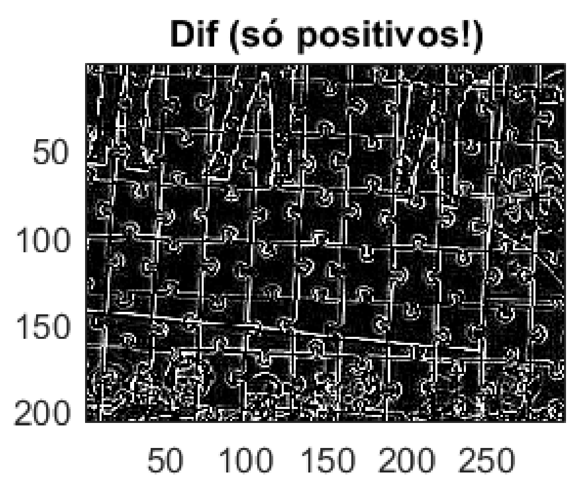
Diferença entre o central e seus vizinhos

```
-1 0 0   0 -1 0   0  0 -1
 0 1 0   0  1 0   0  1  0
 0 0 0   0  0 0   0  0  0

 0 0 0   0 0 0    0  0  0
-1 1 0   0 0 0    0  1 -1
 0 0 0   0 0 0    0  0  0

 0 0 0   0  0 0   0  0  0
 0 1 0   0  1 0   0  1  0
-1 0 0   0 -1 0   0  0 -1
```

```
-1 -1 -1
-1  8 -1
-1 -1 -1
```

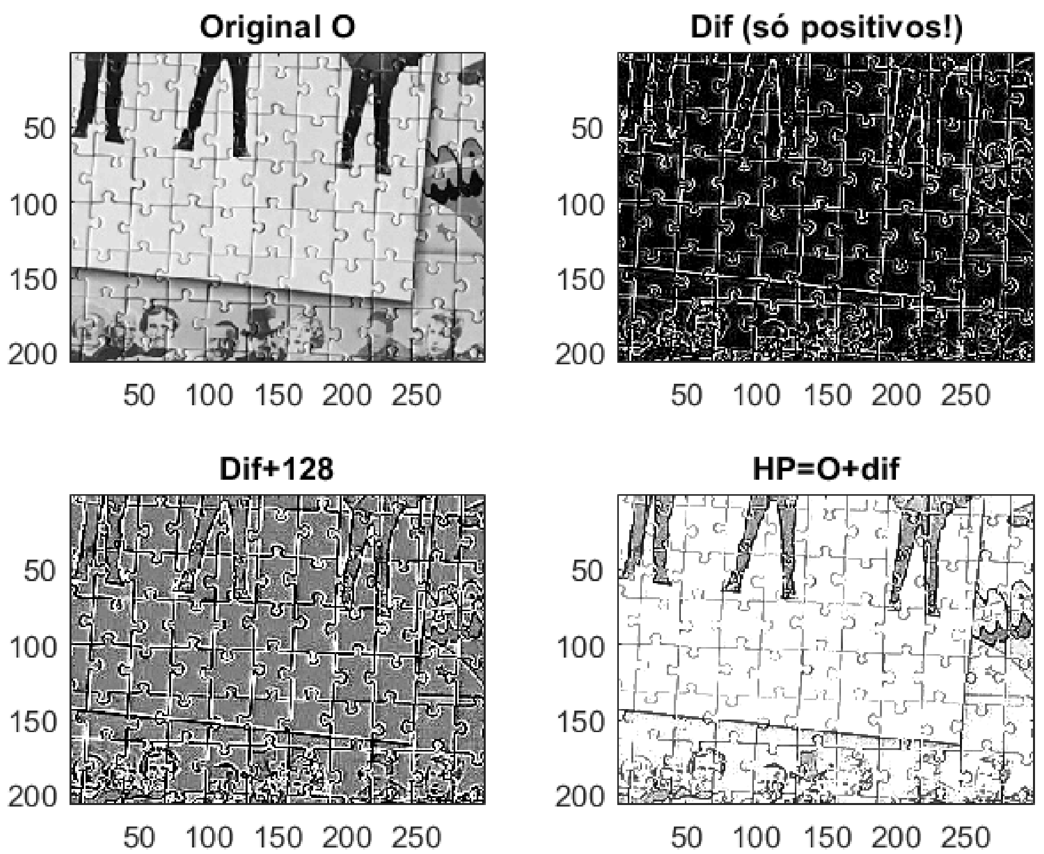
A diferença entre o central e seus oito vizinhos





Diferença + Central

$$\begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix} \quad + \quad \begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix} \quad \Rightarrow \quad \begin{matrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{matrix}$$



Somando a diferença ao valor central acentua o contraste

Exemplos de passa-altas

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 25 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

Passa altas

- Original

passa-altas

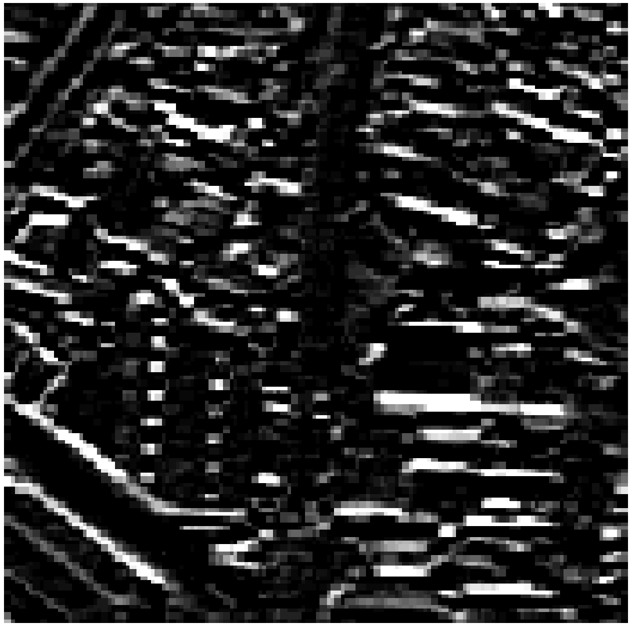
Filtros direcionais

- A convolução de uma janela e a imagem também é útil para salientar determinadas linhas ou bordas. Por exemplo, as técnicas de filtragem permitem salientar as bordas ou linhas que ocorrem numa determinada direção, fazendo a diferença dos valores na janela considerando sua posição em relação ao pixel central da janela. A seguir são mostrados alguns exemplos destes filtros.

Exemplo: bordas horizontais

1	1	1
1	-2	1
-1	-1	-1

Norte



- Os contrates na direção norte são salientados.
- Algumas linhas diagonais também são salientadas, pois possuem uma componente norte forte.

$$\begin{array}{|rrr|}\hline 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \\ \hline\end{array}$$

Norte

$$\begin{array}{|rrr|}\hline 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \\ \hline\end{array}$$

Nordeste

$$\begin{array}{|rrr|}\hline -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \\ \hline\end{array}$$

Sul

$$\begin{array}{|rrr|}\hline 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \\ \hline\end{array}$$

Noroeste

$$\begin{array}{|rrr|}\hline -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \\ \hline\end{array}$$

Leste

$$\begin{array}{|rrr|}\hline -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \\ \hline\end{array}$$

Sudeste

$$\begin{array}{|rrr|}\hline 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \\ \hline\end{array}$$

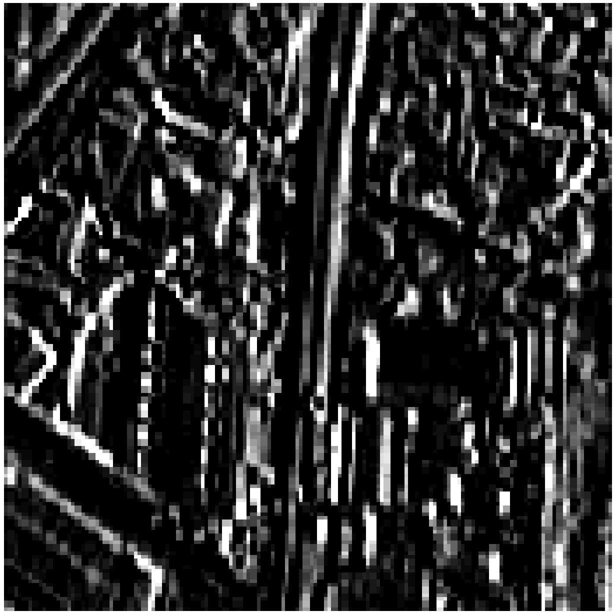
Oeste

$$\begin{array}{|rrr|}\hline 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \\ \hline\end{array}$$

Sudoeste

-1	1	1
-1	-2	1
-1	1	1

leste



Filtros não lineares

- Resultam da análise da vizinhança em torno do pixel, mas neste caso seu funcionamento não pode ser representado usando a forma geral da convolução.
- **O filtro de moda** (valor mais frequente) : Usado para suavizar imagens, especialmente temáticas, pois o novo valor atribuído ao pixel central corresponde ao valor mais frequente da vizinhança e por este motivo é igual a pelo menos um dos pixels vizinhos.
- **O filtro de mediana** (valor central) : o novo valor corresponde ao valor central após ordenar os valores de forma crescente.

1	1	1	1	2	1	1	0
1	1	2	1	3	1	1	1
1	1	3	1	2	1	1	0
1	1	3	1	4	1	1	1
9	7	8	8	8	9	9	8
9	9	8	8	9	9	9	8
9	9	9	8	9	8	9	9
9	8	9	8	8	9	9	9

Serie=[3 1 2 3 1 4 8 8 8]
Moda: 8 (mais frequente)

Mediana:
[1 1 2 3 3 4 8 8 8] = 3

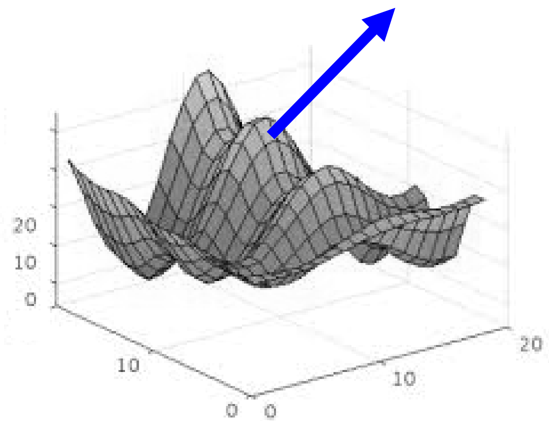
Mediana

Filtro da mediana: O valor resultante é a mediana da vizinhança. Este filtro introduz um certo grau de suavização na imagem resultante, do que decorre perda de detalhe. A diferença em relação ao filtro passa baixas é que as bordas não são degradadas em extremo, pois os valores originais são preservados.

Filtros de Gradiente

O gradiente de uma superfície descreve sua inclinação no local especificado e é um vetor, que aponta para fora da superfície.

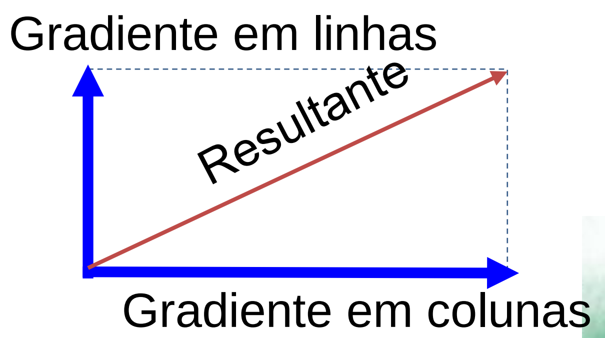
O gradiente pode ser calculado a partir de suas duas componentes (Norte e Leste), ou seja, a derivada parcial da função da superfície em relação a linhas e colunas.



$$G(x,y) = \begin{bmatrix} \dfrac{\delta F(x,y)}{\delta(x)} \\ \dfrac{\delta F(x,y)}{\delta(y)} \end{bmatrix}$$

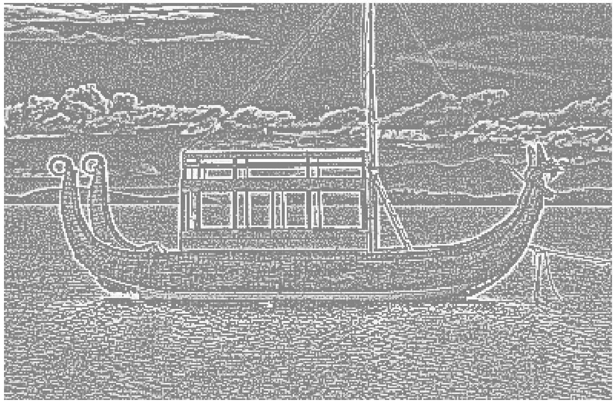
Intensidade do Gradiente

$$I(x,y) = \sqrt{\frac{\delta F(x,y)}{\delta(x)}^2 + \frac{\delta F(x,y)}{\delta(y)}^2}$$



No Para calcular o Gradiente:

- Estima-se o gradiente em X
- Estima-se o gradiente em Y (Y perpendicular a X)
- Calcula-se a resultante da soma destes dois vetores.
- O pixel recebe um valor proporcional à magnitude do gradiente.



No processamento de imagens, pode-se assumir que a variação dos valores digitais se assemelha a uma superfície, similar a um Modelo Digital do Terreno (MDT).

Logo, torna-se possível calcular o gradiente para qualquer pixel, analisando a variação dos valores em sua vizinhança.

Para isto:

- Estima-se o gradiente em X
- Estima-se o gradiente em Y (Y perpendicular a X)
- Calcula-se a resultante da soma destes dois vetores.
- O pixel recebe um valor proporcional à magnitude do gradiente.

A diferença entre os filtros de gradiente radica na maneira de estimar as duas derivadas parciais.

Ex: filtros de Roberts:

$$\boxed{\begin{matrix} 1 & 0 \\ 0 & -1 \end{matrix}} \qquad \boxed{\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}}$$

$\delta F(x,y)/\delta(x)$ $\quad$ $\delta F(x,y)/\delta(y)$

Magnitude: $$I(x,y) = \sqrt{\frac{\delta F(x,y)}{\delta(x)}^2 + \frac{\delta F(x,y)}{\delta(y)}^2}$$

$\delta F(x,y)/\delta(x) = F(x\text{-}1,\ y\text{-}1) - F(x,y)$

$\delta F(x,y)/\delta(y) = F(x\text{-}1,\ y+1) - F(x,y)$

- Ex: filtros de Prewitt

$$\begin{array}{ccc} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{array} \qquad \delta F(x,y)/\delta(x)$$

$$\begin{array}{ccc} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{array} \qquad \delta F(x,y)/\delta(y)$$

Magnitude:

$$I(x,y) = \sqrt{\frac{\delta F(x,y)}{\delta(x)}^2 + \frac{\delta F(x,y)}{\delta(y)}^2}$$

- Ex: filtros de Sobel

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

δF(x,y)/δ(x)

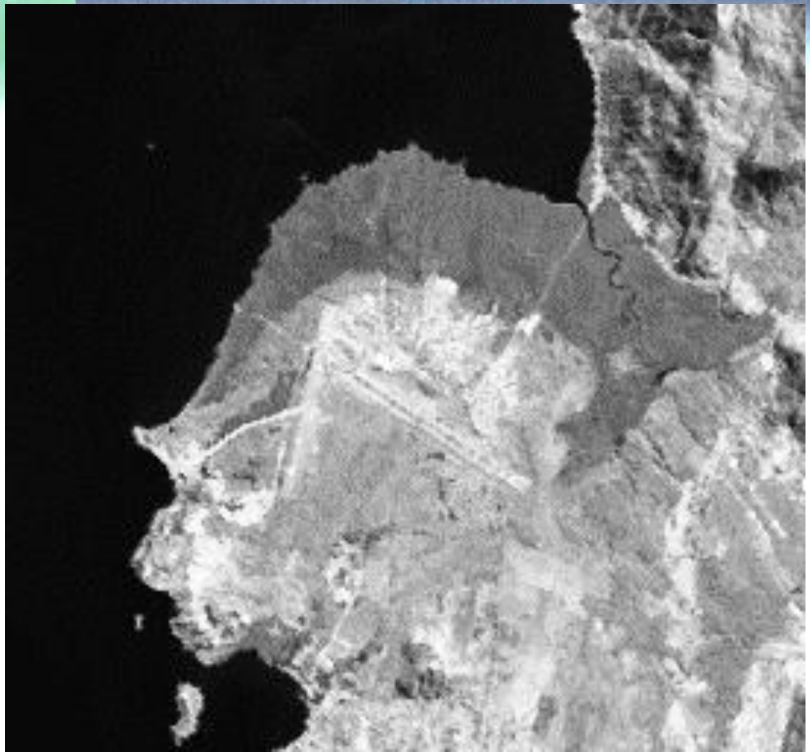
$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

δF(x,y)/δ(y)

Magnitude:

$$I(x,y) = \sqrt{\frac{\delta F(x,y)}{\delta(x)}^2 + \frac{\delta F(x,y)}{\delta(y)}^2}$$

Ex: filtro de SOBEL



Áreas uniformes: baixo gradiente
áreas de fronteira são salientadas, bem como feições lineares.

Laplaciano

O filtro Laplaciano é um operador que calcula a derivada isotrópica (não depende da direção, em tidas as direções)

Gradiente local em todas as direções

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Exemplo:

0	1	0
1	-4	1
0	1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

Na prática, o Laplaciano pode ser muito demorado para calcular e é sensível à presença de ruído. Por isso, não se usa diretamente sua formulação original.

Usa-se a diferença entre a imagem original e a imagem suavizada com um filtro Gaussiano.

Isto é conhecido como o Laplaciano do Gaussiano

LoG

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1-\frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- Como seria um programa de filtro?

dtype = None

```
img = cv2.imread(nome)
n,m,nb = img.shape
I=img[:,:,2]
J=np.zeros([n,m], dtype='float')
for i in range(n):
  for j in range(m):
     S=0;
     for dl in range(3):
             for coluna in range(3):
                   s=s+I[i+dl-1, j+dc-1]
       s=s/9
       J(i,j)=s
cv2_imshow(J)
```

```
1 1 1 1 2 1 1 0
1 1 2 1 3 1 1 1
1 1 3 9 2 1 1 0
1 1 3 1 2 1 1 1
9 7 8 9 8 9 9 8
9 9 8 8 9 9 9 8
9 9 9 8 9 8 9 9
9 8 9 8 8 9 9 9
```

Isto não funciona para os pixels nas bordas da imagem, pois o vizinho anterior à primeira linha não existe!

Obs: o que ocorre se somarmos os oito valores uint8 ?

Restringir pixels a serem processados em função do tamanho da vizinhança.

```
I=img[:,:,2]
J=np.zeros((n,m),dtype = np.uint8 )  # criamos uma variável vazia em uint8
for i in range(1,n-1):
  for j in range(1,m-1):
    s=0
    for dl in range(3):
      for dc in range(3):
        v=float( I[i+dl-1, j+dc-1]) # converter a float para somar mais de 255
        s=s + v
    s=np.round(s/9)
    J[i,j]=np.uint8(s)
cv2_imshow(J)
```

1	1	1	1	2	1	1	0
1	1	2	1	3	1	1	1
1	1	3	9	2	1	1	0
1	1	3	1	2	1	1	1
9	7	8	9	8	9	9	8
9	9	8	8	9	9	9	8
9	9	9	8	9	8	9	9
9	8	9	8	8	9	9	9

- No ambiente Google Colab

- Desenvolva um filtro de média (3x3), depois um (5x5)
- Um filtro passa-altas...

Programa (Passa-baixas)

1	1	1
1	1	1
1	1	1

1	1	1	1	2	1	1	0
1	1	2	1	3	1	1	1
1	1	3	9	2	1	1	0
1	1	3	1	2	1	1	1
9	7	8	9	8	9	9	8
9	9	8	8	9	9	9	8
9	9	9	8	9	8	9	9
9	8	9	8	8	9	9	9

Elabore um programa de filtro de médias (passa-baixas) no que o usuário possa variar o tamanho da janela móvel)

dim=3,5,7,..?

Vizinhos antes e depois

lado=(dim-1)/2

Verifique para dim=3, ou 5

No 3x3, a varredura não pode ser feita para o primeiro pixel, devemos começar no elemento (lado+1)

e não podemos terminar na ultima linha (n) mas devemos terminar em (n-[lado-1])

E no caso 5x5 ?

1	1	1	1	2	1	1	0
1	1	2	1	3	1	1	1
1	1	3	9	2	1	1	0
1	1	3	1	2	1	1	1
9	7	8	9	8	9	9	8
9	9	8	8	9	9	9	8
9	9	9	8	9	8	9	9
9	8	9	8	8	9	9	9

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1